11-15-04

AF

DOCKET NO.: TN222/USYS-0083 PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In Re Application of:
Karl Wilmer Scholtz, James S. Irwin and Samir Tamri

Confirmation No.: 1300

Application No.: 09/702,224

Group Art Unit: 2128

Filing Date: October 31, 2000

Examiner: Thai Q. Phan

For: DIALOGUE FLOW INTERPRETER DEVELOPMENT TOOL

EXPRESS MAIL LABEL NO: EV 392215254 US
DATE OF DEPOSIT: November 12, 2004

EV302215254US

MS Appeal Brief - Patent
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF TRANSMITTAL PURSUANT TO 37 CFR § 41.37

Transmitted herewith is the APPEAL BRIEF in this application with respect to the Notice of Appeal received by The United States Patent and Trademark Office on **September 13, 2004.** Also transmitted herewith are copies of two references cited in the Appeal Brief, U.S. Patent No. 6,246,981 B1 and U.S. Patent No. 6,513,009 B1.

☐ Applicant(s) has previously claimed small entity status under 37 CFR § 1.27 .

☐ Applicant(s) by its/their undersigned attorney, claims small entity status under 37 CFR § 1.27 as:

- ☐ an Independent Inventor
- ☐ a Small Business Concern
- ☐ a Nonprofit Organization.

☐ Petition is hereby made under 37 CFR § 1.136(a) (fees: 37 CFR § 1.17(a)(1)-(4) to extend the time for response to the Office Action of to and through comprising an extension of the shortened statutory period of month(s).

	SMALL ENTITY		NOT SMALL ENTITY	
	RATE	FEE	RATE	FEE
☒ **APPEAL BRIEF FEE**	$170	$	**$340**	**$340.00**
☐ ONE MONTH EXTENSION OF TIME	$55	$	$110	$
☐ TWO MONTH EXTENSION OF TIME	$215	$	$430	$
☐ THREE MONTH EXTENSION OF TIME	$490	$	$980	$
☐ FOUR MONTH EXTENSION OF TIME	$765	$	$1530	$
☐ FIVE MONTH EXTENSION OF TIME	$1040	$	$2080	$
☐ LESS ANY EXTENSION FEE ALREADY PAID	minus	($)	minus	($)
TOTAL FEE DUE		$0		**$340.00**

☐ A check in the amount of **$ .00** is attached. Please charge any deficiency or credit any overpayment to Deposit Account No. 23-3050.

☒ Please charge Deposit Account No. 23-3050 in the amount of **$340.00**. This sheet is attached in duplicate.

☒ The Commissioner is hereby requested to grant an extension of time for the appropriate length of time, should one be necessary, in connection with this filing or any future filing submitted to the U.S. Patent and Trademark Office in the above-identified application during the pendency of this application. The Commissioner is further authorized to charge any fees related to any such extension of time to deposit account 23-3050. This sheet is provided in duplicate.

Date: November 12, 2004

Nathaniel Ari Long
Registration No. 53,233

Woodcock Washburn LLP
One Liberty Place - 46th Floor
Philadelphia PA 19103
Telephone: (215) 568-3100
Facsimile: (215) 568-3439

DOCKET NO.: TN222 (USYS-0083) PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of: Confirmation No.: **1300**

Karl Wilmer Scholz, James S. Irwin, and Samir Tamri

Serial No.: **09/702,224** Group Art Unit: **2128**

Filing Date: **October 31, 2000** Examiner: **Phan, Thai Q.**

For: **DIALOGUE FLOW INTERPRETER DEVELOPMENT TOOL**

EXPRESS MAIL LABEL NO: EV 302215254 US
DATE OF DEPOSIT: November 12, 2004

Mail Stop Appeal-Brief Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

APPEAL BRIEF PURSUANT TO 37 C.F.R. § 41.37

This brief is being filed in support of Appellant's appeal from the rejections of claims 1-11 dated March 12, 2004. A Notice of Appeal was filed on September 13, 2004.

11/16/2004 JBALINAN 00000145 233050 09702224
01 FC:1402 340.00 DA

REAL PARTY IN INTEREST

UNISYS CORPORATION is the real party in interest in the present application. The inventors in the present application assigned their interest to UNISYS CORPORATION on March 15, 2001 as recorded by the United States Patent and Trademark Office ("USPTO") on March 23, 2001 at reel 011655, frame 0707.

RELATED APPEALS AND INTERFERENCES

There are no related appeals or interferences.

STATUS OF CLAIMS

Claims 1-11 stand rejected under 35 U.S.C. 103(a) as being directed to subject matter that allegedly would have been obvious over U.S. Pat. 6,513,009 B1 ("Comerford") in view of U.S. Pat. 6,246,981 B1 ("Papineni"). The rejections of claims 1-11 are appealed.

STATUS OF AMENDMENTS

A listing of claims is provided in the claims appendix, below. Claims that have never been amended (namely, claims 2-4, 6, 7, 9, and 11) are marked as "original." Claims that have been modified by entered amendments (namely, claims 1, 5, and 10) are marked as "Previously Presented." Claims that have been amended after final (namely, claim 8), where the amendments have not yet been entered, are marked as "Currently Amended." Please note that, pursuant to Rule 41.37(c)(2), the amendments made after final and not admitted are not included in the claims appendix or argued as allowable due to the amended features. The proposed but unentered amendments to claim 8 are underlined below:

8. (Currently Amended): A computer-readable medium comprising computer executable instructions for instructing a computer to perform the acts of:

accepting instructions into a design tool, said instructions specifying a programmer-defined flow of conversation between a human and a computer;

said design tool producing a data file for input to an interpreter;

interpreting said data file; and

providing the human-computer dialogue-enabled interaction.

SUMMARY OF CLAIMED SUBJECT MATTER

Independent Claim 1

The invention is directed to voice interaction between a human and a computer. For example, when a person places a telephone call to customer service departments of many large companies, he or she may interact with a computer that accepts spoken inputs. Most modern consumers are familiar with systems that ask, "Please say your account number." Independent claim 1 is directed to *the creation of* such applications.

The claims, including claim 1, thus contain features that are particular to a development/design tool for voice-enabled applications. Applicants appeal the outstanding rejections because the cited art does not provide a design tool, and therefore does not combine a design tool with the various other features of the invention reflected in the claims.

Claim 1 provides:

1. A method of developing a dialogue-enabled application for executing on a computer that enables a human and a computer to interact, comprising the acts of:

(a) inputting instructions specifying the flow of a conversation to a design tool, said design tool producing a data file, said data file containing information relating to prompts, responses, branches and conversation flow for implementing a programmer-defined human-computer speech-enable interaction; and

(b) instantiating an interpreter object within an application, the interpreter object interpreting the data file to provide the programmer-defined human-computer dialogue-enabled interaction defined by the data file.

The systems and methods provided generally contemplate an arrangement such as that of App. Fig. 2. Fig. 2 shows a software application 230 that incorporates aspects of the invention. These aspects can be introduced and understood with reference to four basic components in Fig. 2. A first component, the "Voice Input / Output Device," accepts a spoken input, and produces a digital output. A second component, the "Recognition Engine," converts the digital data to machine readable text. The third component, or "Language Interpreter" creates a token representation of the meaning of the user input. The final component, or "Dialog Flow Interpreter (DFI)" 232 controls call flow by proceeding as necessary based on the token response from the Language Interpreter.

To illustrate the above, imagine a system in which a caller is asked by an automated system, "Please say 'yes' or 'no.'" In response, the caller says "Clearly, my answer is yes." The caller has a thick accent and slurs his words. First, the caller's response would be converted to a digital signal by the Voice I/O Device. Next, the Recognition Engine would determine, despite the accent and the slur, that the response was, "Clearly, my answer is yes." The Recognition Engine would produce a machine-readable representation of these words. Next, the Language Interpreter would examine the response, perhaps by looking for a "yes" or a "no" anywhere in the machine-readable representation, and determine that the caller's response was "yes." A token representing that the response was "yes" can be passed to the DFI 232. The DFI 232 would then determine a next action can be taken based on the response. For example, a next question can be asked.

Claim 1 provides "A method of developing a dialog-enabled application" such as application 230. The method comprises "inputting instructions specifying the flow of a conversation ***to a design tool***, said design tool producing a data file..." Refer again to Fig. 2, where Design Tool 210 generates (produces) Data File 220. The

specification provides, "the developer [can] specify generalized instructions about the flow of a conversation...to a DFI design tool 210." App. 4:31-5:2. Claim 1 further elaborates on the contents of the data file by stating that it contains information, "relating to prompts, responses, branches and conversation flow for implementing a programmer-defined human-computer speech-enable (sic) interaction."

Next, the method of claim 1 comprises "instantiating an interpreter object within an application, the interpreter object interpreting the data file to provide the programmer-defined human-computer dialogue-enabled interaction defined by the data file." This step can be understood with reference to Fig. 2, where application 230 includes Dialog Flow Interpreter (DFI) 232. Please also refer to Fig. 5 and corresponding text. Fig. 5 shows an application 510, a data file 520, and an interpreter object 530. The specification provides:

> [T]he software application, 510, coded by the developer in any of a variety of programming languages, instantiates the dialogue flow interpreter, 530, and tells it to interpret the design specified in the file, 520, generated above by the DFI design tool. The dialogue flow interpreter, 530, controls the flow through the application, supplying all the underlying code, 540, that previously the developer would have had to write.

App. 7:17-7:22.

In summary, claim 1 teaches developing a dialogue-enabled application by inputting instructions specifying the flow of a conversation to a design tool. The design tool, e.g. 210 produces a data file, e.g. 220 representative of the conversation. Then, the application, e.g. 230, can conduct a voice interaction by instantiating an interpreter object, e.g. 232, that interprets the data file 220.

Independent Claim 5

Claim 5 provides:

> 5. A system for developing dialogue-enabled software for executing on a computer that enables a human and a computer to interact comprising:
> a design tool for accepting instructions specifying a programmer-defined flow of a conversation, said design tool producing a data file; and

> an interpreter for interpreting said data file, said interpreter automatically enabling the programmer-defined human-computer interaction.

Claim 5 is a system claim that is similar in many respects to claim 1, described above. While claim 1 provides a "method of developing a dialogue-enabled application," claim 5 provides a "system for developing dialogue-enabled software." Thus, claim 5 provides a system for developing software that can engage in voice interactions between a human and a computer.

The system for creating software of claim 5 comprises, first, "***a design tool*** for accepting instructions specifying a programmer-defined flow of a conversation, said design tool producing a data file." Refer to Fig. 2, where Design Tool 210 generates (produces) Data File 220. The specification provides, "the developer [can] specify generalized instructions about the flow of a conversation...to a DFI design tool 210." App. 4:31-5:2. Applicant submits that this discloses a design tool that accepts instructions, because where the developer can specify instructions to a design tool, the design tool must be accepting the instructions. Also, refer to Fig. 4 which shows a process beginning with a designer entering a design for an application 410, and ending with a design tool 400 that generates a file 450 representing the design.

The system of claim 5 next comprises, "an interpreter for interpreting said data file, said interpreter automatically enabling the programmer-defined human-computer interaction." This component of the claimed system can be understood with reference to Fig. 5 and corresponding text. Fig. 5 shows an application 510, a data file 520, and an interpreter object 530. The specification provides:

> [T]he software application, 510, coded by the developer in any of a variety of programming languages, instantiates the dialogue flow interpreter, 530, and tells it to interpret the design specified in the file, 520, generated above by the DFI design tool. The dialogue flow interpreter, 530, controls the flow through the application, supplying all the underlying code, 540, that previously the developer would have had to write.

App. 7:17-7:22. Thus, when the interpreter is instantiated by the developer's application, it "automatically enables the programmer-defined human-computer interaction," by calling on the data file that the developer created using the design tool.

In summary, claim 5 is directed to a system for developing dialogue-enabled software comprising a design tool that can accept instructions specifying the flow of a conversation. The design tool produces a data file representative of the conversation. Then, the system uses an interpreter that can conduct a voice interaction using the data file.

Independent Claim 8

Claim 8 provides the following. Note that unentered amendments after final are not included below:

> 8. A computer-readable medium comprising computer executable instructions for instructing a computer to perform the acts of:
> accepting instructions, said instructions specifying a programmer-defined flow of conversation between a human and a computer;
> producing a data file for input to an interpreter;
> interpreting said data file; and
> providing the human-computer dialogue-enabled interaction.

Claim 8 provides a computer readable medium comprising instructions for a computer. Computer readable media, such as CD-ROMs, floppy disks, computer hard drives and the like are widely used and understood, and are generally referred to at App. 13:12-14:11.

The computer readable medium of claim 8 first comprises instructions for "accepting instructions, said instructions specifying a ***programmer-defined*** flow of conversation between a human and a computer." The specification elaborates that "the developer [can] specify generalized instructions about the flow of a conversation...***to a DFI design tool*** 210, accessible through a programmer-friendly graphical interface." App. 4:31-5:3. For, example, an application that produces a graphical interface can be stored on a computer readable medium, such as a CD-ROM. The graphical interface may provide a vehicle for accepting instructions, such as a "programmer-defined flow of conversation" according to this first aspect of claim 8.

Next, claim 8 provides instructions for "producing a data file for input to an interpreter." Please refer to Fig. 2, where Design Tool 210 generates (produces) Data File 220. As stated in the corresponding description, "[t]he DFI design tool, 210, produces a data file, 220." Referring to the example above, the graphical interface that accepts instructions in accordance with the first part of claim 8 can represent the design tool that subsequently produces a data file in accordance with this second part of claim 8.

Third, claim 8 recites "interpreting said data file." Again referring to Fig. 2 and the corresponding text:

> When the calling program (speech application), 230, which can be written by the developer in a variety of programming languages, executes, the calling program, 230, instantiates the dialogue flow interpreter, 232, providing to the interpreter, 232, the data file, 220, produced by the DFI design tool, 210.

App. 5:4-5:7. Thus, the data file 220 is provided to an interpreter 232 that can be instantiated by a calling application. Claim 8 contemplates that instructions representing such an interpreter 232 for carrying out a voice interaction in accordance with the data file 220 are included on the computer readable medium, along with the instructions for the first two elements of the claim, set forth above.

Finally, claim 8 recites instructions for "providing the human-computer dialogue-enabled interaction. Referring to Fig. 2, the four basic components of a system that provides such a human-computer dialogue-enabled interaction are illustrated. As stated above in the section describing claim 1, a first component, the "Voice Input / Output Device," accepts a spoken input, and produces a digital output. A second component, the "Recognition Engine," converts the digital data to machine readable text. The third component, or "Language Interpreter" creates a token representation of the meaning of the user input. The final component, or "Dialog Flow Interpreter (DFI)" 232 controls call flow by proceeding as necessary based on the token response from the Language Interpreter. Any of these four components, or portions thereof, when stored as instructions on a computer readable medium, are sufficient to meet this final element of claim 8.

In summary, claim 8 is directed to a computer readable medium with instructions for, first, accepting developer instructions, and second, producing a data file. These first two elements of claim 8 are met, for example, by a design tool with a user interface for entry of instructions and a process for converting the instructions into a data file. Next, claim eight teaches interpreting the data file and providing a human-computer dialog enabled interaction. These last two elements are met by an "interpreter" component that can be instantiated by an application, and directed to the created data file to perform the desired voice interaction.

Independent Claim 10

Claim 10 provides:

> 10. A dialogue flow interpreter (DFI) for use in computer-implemented system for carrying out a dialogue between a human and a computer, wherein the DFI comprises computer executable instructions for reading a data file containing programmer-predefined information concerning prompts, responses, branches and conversation flow for implementing a human-computer dialogue, and computer executable code for using said information in combination with a library of shared objects to conduct said dialogue.

Claim 10 can be properly understood with reference to Fig. 2, Fig. 5, and corresponding text. First, claim 10 refers to "[a] dialogue flow interpreter (DFI) ... for carrying out a dialogue between a human and a computer." Fig. 2 provides such a DFI 232, as does Fig. 5—element 530. The DFI is described in the specification as a functional unit that can draw on a customized data file, or DFI object, to produce a voice interaction. *See* App. 5:6-5:10.

Claim 10 itself proceeds to further describe the claimed DFI by specifying that it comprises instructions for "reading a data file ***containing programmer-predefined information***." The claim elaborates that the information concerns, "prompts, responses, branches and conversation flow for implementing a human-computer dialogue." Fig. 2 illustrates this aspect of the claim by showing a data file 220 that is "used by" the DFI 232. Moreover, Fig. 2 provides a design tool 210 that "generates" the data file 220. The specification provides, "the developer [can] specify generalized instructions about the

flow of a conversation...***to a DFI design tool*** 210." App. 4:31-5:2. Thus, while the design tool 210 is not expressly required by claim 10, the claim provides a DFI, e.g. 232, that includes instructions for reading a data file that contains programmer-predefined information, such as 220.

Finally, claim 10 recites that the DFI includes instructions for "using [the data file] in combination with a library of shared objects to conduct said dialogue." The DFI 530 of Fig. 5 draws upon both a data file 520 and a library of shared objects 540. As described in the specification, "[t]he Dialogue Flow Interpreter, or DFI, of the present invention provides a library of "standardized" objects that implement low-level details of dialogues." App. 7:28-7:29. Thus, in summary, claim 10 is directed to a DFI that can take custom instructions represented in a data file and carry out voice dialog using the various functions provided by a library of shared objects.

Claim 10 thus reaches the novel DFI that permits easy creation of voice dialog software without also requiring the design tool that is described in the specification and that may be provided along with the DFI in many embodiments. This is important from Applicants' the perspective because claim 10 thus reaches infringers who use software that incorporates the invented DFI as well as those who make software using the design tool/novel DFI combination.

GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Claims 1-11 were rejected under 35 U.S.C. 103(a) as allegedly obvious over U.S. Pat. 6,513,009 B1 ("Comerford") in view of U.S. Pat. 6,246,981 B1 ("Papineni").

ARGUMENT

Rejection of Claims 1-11 Under 35 U.S.C. 103(a)

1. Claims 1, 2, and 4

Applicants respectfully submit that the rejection of claims 1, 2, and 4 was incorrect. Rejections under 35 U.S.C. 103(a) require that the reference (or combination of references) disclose every element of the claim. As stated in the M.P.E.P., "the prior art references (or references when combined) must teach or suggest all the claim limitations." M.P.E.P. § 706.02(j). Also, "[o]bviousness can only be established by combining or modifying the teachings of the prior art to produce the claimed invention where there is some teaching, suggestion, or motivation to do so found either explicitly or implicitly in the references themselves or in the knowledge generally available to one of ordinary skill in the art." M.P.E.P. 2143.01.

Furthermore, for a proper rejection under 35 U.S.C. 103(a), "the prior art must provide a motivation or reason for the worker in the art, *without the benefit of [applicant's] specification*, to make the necessary changes in the reference device." M.P.E.P. § 2144.04 (citing *Ex parte Chicago Rawhide Manufacturing Co.*, 223 U.S.P.Q. 351, 353 (Bd. Pat. App. & Inter. 1984) (emphasis added). To establish a prima facie case of obviousness, "there must be some teaching, suggestion or motivation in the prior art to make the specific combination that was made by the applicant." *In re Dance*, 160 F.3d 1339, 1343 (Fed. Cir. 1998). "In other words, the examiner must show reasons that the skilled artisan, confronted with the same problem as the inventor and with no knowledge of the claimed invention, would select the elements from the cited prior art references for combination in the manner claimed." *In re Rouffet*, 149 F.3d 1350, 1357 (Fed. Cir. 1998).

Here, Applicants respectfully assert that the Examiner has failed to establish a *prima facie* case of obviousness because neither Comerford nor Papineni provide specific guidance that would lead one of ordinary skill in the art to the invention of claims 1, 2, or 4. More particularly, and as discussed below, the combination of Comerford and Papineni proposed by the Examiner, assuming only for the sake of argument that such a combination is proper, fails to teach or suggest the combination of elements provided in claim 1. This is because **claim 1 is directed to a method that makes use of a development/design tool by programmers**, subject matter on which Comerford and Papineni are silent.

In the Official Action of March 12, 2004, the Examiner seems to have taken the position that the primary reference, Comerford, discloses all elements of Applicants' claims 1, 2, and 4 except for the feature of "programmer defined". *See* Official Action, pages 2-3. Specifically, the Examiner admits that Comerford fails to meet the "programmer defined" limitation, Official Action, page 3, lines 7-9, but has taken the position that Papineni discloses this. Official Action, page 3, lines 9-12. Applicants respectfully disagree with both of the above assertions—Comerford does not disclose all the elements of claim 1 except "programmer-defined," and Papineni does not disclose "programmer-defined" or otherwise resolve the deficiency of Comerford..

Comerford does not teach or suggest "inputting instructions specifying the flow of a conversation ***to a design tool***," as provided by claim 1, and as incorporated into claims 2 and 4 by virtue of their depending from claim 1. Instead, as Comerford's title suggests, Comerford discloses a "low resource" dialog manager. The Office Action is not clear but *may* imply that Comerford's "user interface" is a design tool that allows for "inputting instructions specifying the flow of a conversation to a design tool" as recited in Applicants' claims. Official Action, page 3, lines 6-7. However, this implication is not supported by the reference.

The user interface in Comerford is illustrated in Comerford Fig. 1B, element 1300, Comerford 6:44-6:55, Comerford 7:8-7:20, and Comerford 7:30-7:48. Comerford's Fig. 1B illustrates several flavors of user interface tables, including SLI Data 1320, Launch Data 1330, and Application Data 1340. Fig. 1B does not illustrate any component for the *design* of 1320, 1330, or 1340, nor does it show a component within 1300 that suggests a *design tool* function.

Comerford 6:44-6:55 provides "four kinds of user interface tables," but assuming for the sake of argument, and contrary to the facts, that Comerford's UI table is analogous to the "data file" of the present invention, Comerford nonetheless makes no mention of how those user interface tables are designed, much less a design tool for designing such tables. Comerford 7:8-7:20 discusses initialization of user interface tables, and discloses that the user interface tables can be updated to support new applications by adding at least one user interface file. Again, assuming for the sake of argument, and

contrary to the facts, that Comerford's UI table is analogous to the "data file" of the present invention, no mention is made for a design tool used to create such a file.

Comerford 7:30-7:48 discusses the file registration list 1310 for registering the various files in the user interface 1300, and again fails to mention or suggest a design tool for use in conjunction therewith. The same is true of Comerford 9:15-10:12, which discusses various other aspects of Comerford's user interface. Applicants submit that the absence of any reference to a design tool or design tool equivalent is entirely consistent with Comerford's invention, namely, and end user application for speech recognition. While Comerford's invention is updateable, there is no design tool for creating Comerford's updates. Thus, Comerford does not disclose a design tool, because Comerford is not interested in designing new speech-enabled applications, but rather supporting existing speech-enabled applications.

The Official Action may, alternatively, assert that Comerford Fig. 1, Comerford 8:38-8:64, and Comerford 9:25-9:41 disclose a design tool that allows for "inputting instructions specifying the flow of a conversation to a design tool" as recited in Applicants' claims. Official Action, page 2, last paragraph through page 3 lines 1-2. However, this implication is also not supported by the reference.

Nowhere in Comerford's Fig. 1A or Fig. 1B is a design tool illustrated. Comerford 8:38-8:64 discloses a series of actions taken by Comerford's interpreter element. Significantly, "[t]he interpreter compares the message source with a list of *hard coded message sources* (including hardware, engines, and applications), and, finding the message is from a hardware system, branches the hardware system code." Comerford 8:46-8:50 (emphasis added). No element for modification of these message sources is provided, and the fact that they are hard coded does not suggest they are modifiable, such as by a design tool.

Comerford 8:54-8:58 states that table driven methods are not essential for initialization because, "all user input to the process may be made through the data file of Fig. 4." Applicants Comerford does not go on to mention the creation of such a data file, much less a design tool for assisting in such creation.

Furthermore, Papineni cannot be used to supplement Comerford's deficiency as a reference against the present invention. Papineni addresses a need for a voice interaction system, or dialog manager, that is more versatile in interacting with a user, can interact with a user on a wide range of topics in natural language, and is easily adaptable to new tasks. *See* Papineni 2:66-3:5. However, the versatility of Papineni's invention should not be mistaken for programmer-defined. A user *interacts* with Papineni's versatile system, but the user does not *modify or design* the system. Thus, Papineni's dialogue management system, like Comerford's invention, is directed to an end user application and not a method or system for designing dialog-enabled applications.

Papineni's specification is substantially directed to selection of a relevant predefined form responsive to the user's input information. As stated in Papineni:

> A task-oriented dialog manager in accordance with the invention performs tasks according to user's request. To perform the tasks, a computer/machine may need many pieces of information. The user may not always supply all pieces of information in one turn of the conversation. The computer/machine needs to query the user for missing information.

Papineni 5:2-5:6. Thus, Papineni is directed to discovering and requesting the information needed to determine what a user needs and then provide the user with the appropriate information. Much of Papineni's specification goes into great detail about how the system would be used to fill out a form. *See*, e.g. Papineni 10:12-10:42. Thus, Papineni gives a powerful dialog-driven application, but ***fails to mention a design tool for a programmer to define the dialog*** for a given application.

Turning to the specific allegations made in the Official Action, it is asserted that "Papineni teaches the feature of programmer defined or preprogrammed in dialog manager in human-computer dialog interactive interface." Official Action 3:10-3:12. Applicants note that the Official Action apparently misunderstands the use of the term "programmer-defined" in claim 1. ***This term is not interchangeable with "preprogrammed."*** Papineni's invention is only "programmer-defined" in a sense that a

programmer writes the code for Papineni's end user application, which is then distributed to users. As used in claim 1, however, the term refers to a programmer that is using the design tool as part of designing a speech-enabled application.

Thus, in the context of claim 1, programmer-defined can be construed to mean "custom," because each human-computer interaction, designed using the design tool of the invention, can be tailored to the needs identified by a programmer. Note that Papineni's system, referring to those portions cited by the Official Action, page 3, lines 12-17, allows for flexible interaction with a user, but the program itself is fixed and not customizable. Programmers are not using Papineni's system to create new dialog-enabled software.

Accordingly, Applicants respectfully submit that the combination of Comerford and Papineni is insufficient to establish a prima facie case of obviousness for claims 1, 2, and 4 under M.P.E.P. § 2143.01. Neither Comerford nor Papineni provide specific guidance that would lead one of ordinary skill in the art to the combination of elements in those claims.

2. Claim 3

Claim 3 includes the limitations of claim 1 that are argued with respect to claims 1, 2, and 4, above. Therefore claim 3 is also considered to define over the Comerford and Papineni references for the reasons given. However, claim 3 is considered independently patentable and is therefore argued here in a separate group from claims 1, 2, and 4.

Claim 3 provides, in its entirety:

> 3. The method of claim 1 wherein said data file is automatically stored.

The specification elaborates on the subject of claim 3 as follows:

> The development tool, 200, automatically saves reusable code of any level of detail, including dialogue objects, in a library that can be made accessible for use in other applications. A dialogue object is a collection of one or more dialogue states including the processing involved in linking the states together.

App. 5:16-5:19. This feature of automatically saving reusable code, including dialog objects, is novel and enhances the potential for reuse of dialogs created using the design tool of claim 1.

The Official Action states that Comerford discloses data files that are automatically stored in Comerford columns 5-13. Official Action, page 4, lines 3-4. Applicants note that this citation spans a large amount of material and a more specific citation would be useful. After a review of this section, Applicants find several references to data files, e.g. Comerford 8:57 and 9:20-9:21. However, these data files are not "automatically stored," because, in Comerford, the data files are preexisting and are not created in the sense that the data files of claim 3 are created by a design tool.

Therefore, first, the data files of Comerford are not produced by a design tool, as provided in claim 1. Second, Comerford's data files are not "automatically stored" as in claim 3, but are rather loaded into active memory when Comerford's end user application initializes. Thus, Comerford provides:

> The hardware system code continues by loading the remainder of the files in the User Interface Files data collection into conventional data structures.

Comerford 9:17-9:20. This description indicates ***a loading of preexisting User Interface Files into active memory*** data structures. It is the initialization of files, not the storage of files. Applicants maintain that the automatic *storage* of data files, once the files are created by the design tool, as in claim 3, is not analogous to initialization as disclosed in Comerford.

Accordingly, applicants respectfully submit that the combination of Comerford and Papineni is insufficient to establish a prima facie case of obviousness for claim 3 under M.P.E.P. § 2143.01. Neither Comerford nor Papineni provide specific guidance that would lead one of ordinary skill in the art to the combination of elements in claim 3.

3. Claims 5 and 7

Claims 5 and 7 are so-called "system" claims that include many similar limitations to claim 1. Claims 5 and 7 contain several differences in scope and novelty, by virtue of being system rather than method claims, and therefore are considered independently patentable. Here, Applicants respectfully assert that claims 5 and 7 are directed to a combination of a ***design tool*** that produces a data file and ***an interpreter*** for enabling a human-computer interaction, and this combination is not disclosed or suggested by Comerford, Papineni, or the intersection thereof.

In the Official Action of March 12, 2004, the Examiner seems to have taken the position that the primary reference, Comerford, discloses all elements of Applicants' claims 5 and 7, except for the feature of "programmer defined". *See* Official Action, pages 2-3. Specifically, the Examiner admits that Comerford fails to meet the "programmer defined" limitation, Official Action, page 4, 21-22, but has taken the position that Papineni discloses this. Official Action, page 5, lines 2-3. Applicants respectfully disagree with both of the above assertions—Comerford does not disclose all the elements except "programmer-defined," and Papineni does not disclose "programmer-defined" or otherwise resolve the deficiency of Comerford.

As stated above with reference to claims 1, 2, and 4, Comerford does not teach or suggest a "***design tool*** for accepting instructions specifying a programmer-defined flow of a conversation," as provided by claim 5, and as incorporated into claim 7 by virtue of its depending from claim 5. Please refer to the above section in this brief setting forth reasons for this conclusion with respect to claims 1, 2, and 4. The same considerations and same references to the present specification and Comerford apply.

Furthermore, as with claims 1, 2, and 4, Papineni cannot be used to supplement Comerford's deficiency as a reference against the present invention. The Official Action asserts that "Papineni teaches programmer defined or predefined by programmer in dialog manager in human-computer interaction." Official Action, page 5, lines 2-4. Applicants note that the Official Action apparently misunderstands the use of the term "programmer-defined" in claim 5. ***This term is not interchangeable with "predefined by programmer."*** Papineni's invention is only "programmer-defined" in a sense that a

programmer writes the code for Papineni's end user application, which is then distributed to users. As used in claim 5, however, the term refers to a programmer that is using the design tool as part of designing a speech-enabled application.

Accordingly, Applicants respectfully submit that the combination of Comerford and Papineni is insufficient to establish a prima facie case of obviousness for claim 5 or 7 under M.P.E.P. § 2143.01. Neither Comerford nor Papineni provide specific guidance that would lead one of ordinary skill in the art to the combination of elements in those claims.

4. Claim 6

Claim 6 includes the limitations of claim 5 that are argued with respect to claims 5 and 7, above. Therefore claim 6 is also considered to define over the Comerford and Papineni references for the reasons given. However, claim 6 is considered independently patentable and is therefore argued here in a separate group from claims 5 and 7.

Claim 6 provides, in its entirety:

> 6. The system of claim 5 further comprising a library, wherein the library contains said data files.

The specification elaborates on the subject of claim 6 as follows:

> The development tool, 200, automatically saves reusable code of any level of detail, including dialogue objects, in a library that can be made accessible for use in other applications. A dialogue object is a collection of one or more dialogue states including the processing involved in linking the states together.

App. 5:16-5:19. This feature of automatically saving reusable code including dialog objects in a library is novel and enhances the potential for reuse of dialogs created using the design tool of claim 5.

The Official Action alleges that Comerford discloses data files that are stored in a library in Comerford columns 14-16. Official Action, page 5, lines 14-15. After a review of this section, Applicants find several references to data files, e.g. Comerford 14:13-14:14 ("data sets are stored in files"), Comerford 14:46-14:47 ("user

interface data stored either in files or in data structures"), and Comerford 14:59-14:60 ("adding or removing user interface files" and "adding or removing script files"). However, the data files described are not produced by "a design tool for accepting instructions specifying a programmer-defined flow of a conversation," as required by claim 5. This requirement of claim 5 gives meaning and context to claim 6, because for data files to be both produced by a design tool and contained in a library, the data files must be stored in the library after they are generated by the design tool.

In contrast, the data files of Comerford 14:13-14:14 consist of data created by and used by a playback and recording engine. *See* Comerford 14:7-14:13. In short, recorded sounds such as voices. The data files of Comerford 14:46-14:47 refer to user-interface data. A user interface is generally understood to be a graphical interface for display, in contrast to the data files of claim 6. This is also true of the files referenced in 14:59-14:60.

Accordingly, Applicants respectfully submit that the combination of Comerford and Papineni is insufficient to establish a prima facie case of obviousness for claim 6 under M.P.E.P. § 2143.01. Neither Comerford nor Papineni provide specific guidance that would lead one of ordinary skill in the art to the combination of elements in claim 6.

5. Claims 8 and 9

Claims 8and 9 include many similar limitations to claim 1. Claims 8 and 9 contain several differences in scope and novelty, however, and therefore are considered independently patentable.

Here, Applicants respectfully assert that the Examiner has failed to establish a *prima facie* case of obviousness because claims 8 and 9 are directed to a computer readable medium that contains instructions for a combinations of actions, including, "***accepting instructions, said instructions specifying a programmer-defined flow of conversation*** between a human and a computer." It will be appreciated that this element of claim 8 is an act performed by a design tool or design tool equivalent. An unentered amendment to the claim expressly requiring this feature formerly believed to

be inherent is on record in the file. Thus the arguments applied above in this brief in the section regarding claims 1, 2, and 4 are also relevant here. As such, the elements of claim 8 are not disclosed or suggested by Comerford, Papineni, or the intersection thereof.

The Official Action asserts that "Papineni teaches programmer defined (preprogrammed) in dialog manager for human-computer interaction." Official Action, page 6, lines 14-16. Applicants note that the Official Action apparently misunderstands the use of the term "programmer-defined" in claim 8. ***This term is not interchangeable with "preprogrammed."*** Papineni's invention is only "programmer-defined" in a sense that a programmer writes the code for Papineni's end user application, which is then distributed to users. As used in claim 8, however, the term refers to a programmer that is using the design tool as part of designing a speech-enabled application.

Accordingly, Applicants respectfully submit that the combination of Comerford and Papineni is insufficient to establish a prima facie case of obviousness for claim 8 or 9 under M.P.E.P. § 2143.01. Neither Comerford nor Papineni provide specific guidance that would lead one of ordinary skill in the art to the combination of elements in those claims.

6. Claims 10 and 11

Applicants' claims 10 and 11 are directed to a dialog flow interpreter ("DFI") that uses a data file that comes from the design tool. These claims are considered independently patentable and therefore are argued here as a separate group.

The claimed invention includes a DFI which can work with a "language interpreter" (see claim 11). Thus, there are two "interpreters" that work together. The DFI reads the data file created by the design tool and works in addition to a language interpreter, a recognition engine, and a voice input/output device, as claimed in dependent claim 11.

Again, the Examiner has taken the position that the primary reference, Comerford, discloses all elements except for the feature of "programmer defined" in the dialog manager in a human-computer dialog interactive interface. The Office Action suggests that while Comerford fails to disclose the feature of "programmer defined,"

Papineni teaches the feature of programmer defined or preprogrammed in a dialog manager.

Applicants' again respectfully disagree with the rejection. The "interpreter" of Comerford is perhaps analogous to the "language interpreter" recited in Applicants' claim 11, but there is no corresponding element for the claimed dialog flow interpreter of claim 10. In other words, Comerford's interpreter is not a DFI. Comerford's "interpreter" does not have, and would have no need for, Applicants' claimed design tool and data file features, which work directly with the DFI. Applicants respectfully submit, therefore, that the subject matter of claims 10 and 11 is patentable over Comerford and Papineni, alone or in combination.

Accordingly, Applicants respectfully submit that the combination of Comerford and Papineni is insufficient to establish a prima facie case of obviousness for claim 10 and 11 under M.P.E.P. § 2143.01. Neither Comerford nor Papineni provide specific guidance that would lead one of ordinary skill in the art to the combination of elements in those claims.

Respectfully submitted,

Date: November 12, 2004

Nathaniel Ari Long
Registration No. 53,233

Woodcock Washburn LLP
One Liberty Place - 46th Floor
Philadelphia PA 19103
Telephone: (215) 568-3100
Facsimile: (215) 568-3439

CLAIMS APPENDIX

1. (Previously presented): A method of developing a dialogue-enabled application for executing on a computer that enables a human and a computer to interact, comprising the acts of:

(a) inputting instructions specifying the flow of a conversation to a design tool, said design tool producing a data file, said data file containing information relating to prompts, responses, branches and conversation flow for implementing a programmer-defined human-computer speech-enable interaction; and

(b) instantiating an interpreter object within an application, the interpreter object interpreting the data file to provide the programmer-defined human-computer dialogue-enabled interaction defined by the data file.

2. (Original): The method of claim 1 wherein said data file further contains information concerning a speech recognition engine.

3. (Original): The method of claim 1 wherein said data file is automatically stored.

4. (Original): The method of claim 1 wherein said inputting of instruction takes place through a graphical interface.

5. (Previously presented): A system for developing dialogue-enabled software for executing on a computer that enables a human and a computer to interact comprising:

a design tool for accepting instructions specifying a programmer-defined flow of a conversation, said design tool producing a data file; and

an interpreter for interpreting said data file, said interpreter automatically enabling the programmer-defined human-computer interaction.

6. (Original): The system of claim 5 further comprising a library, wherein the library contains said data files.

7. (Original): The system of claim 5, wherein the design tool further comprises a graphical interface.

8. (Currently amended): A computer-readable medium comprising computer executable instructions for instructing a computer to perform the acts of:

accepting instructions, said instructions specifying a programmer-defined flow of conversation between a human and a computer;

producing a data file for input to an interpreter;

interpreting said data file; and

providing the human-computer dialogue-enabled interaction.

9. (Original): The computer-readable medium of claim 8 containing further instructions enabling the generated code to be immediately accessible to other software developers.

10. (Previously presented): A dialogue flow interpreter (DFI) for use in computer-implemented system for carrying out a dialogue between a human and a computer, wherein the DFI comprises computer executable instructions for reading a data file containing programmer-predefined information concerning prompts, responses, branches and conversation flow for implementing a human-computer dialogue, and computer executable code for using said information in combination with a library of shared objects to conduct said dialogue.

11. (Original): A DFI as recited in claim 10, wherein the DFI is implemented in an application comprising, in addition to the DFI, a language interpreter, recognition engine, and voice input/output device.

EVIDENCE APPENDIX

U.S. Pat. 6,246,981 B1, Papineni et al., first entered by examiner in Official Action dated 8/12/32003.

U.S. Pat. 6,513,009 B1, Comerford et al., first entered by examiner in Official Action dated 10/03/2003.

RELATED PROCEEDINGS APPENDIX

There are no related appeals or interferences.